Big Data in High Energy Physics

Igor Mandrichenko
Draft
5/5/2014

High Energy Physics Data Handling Model

In HEP, data processing is a process of collection, reconstruction and analysis of "event" data recorded by the detector. Event is a minimal unit of data in HEP. Event is a recording of certain physical processes. In HEP, events are not related to each other in any way other than they may be collected under common conditions. Once two events are separated from one another, they can be analyzed completely independently and in any order.

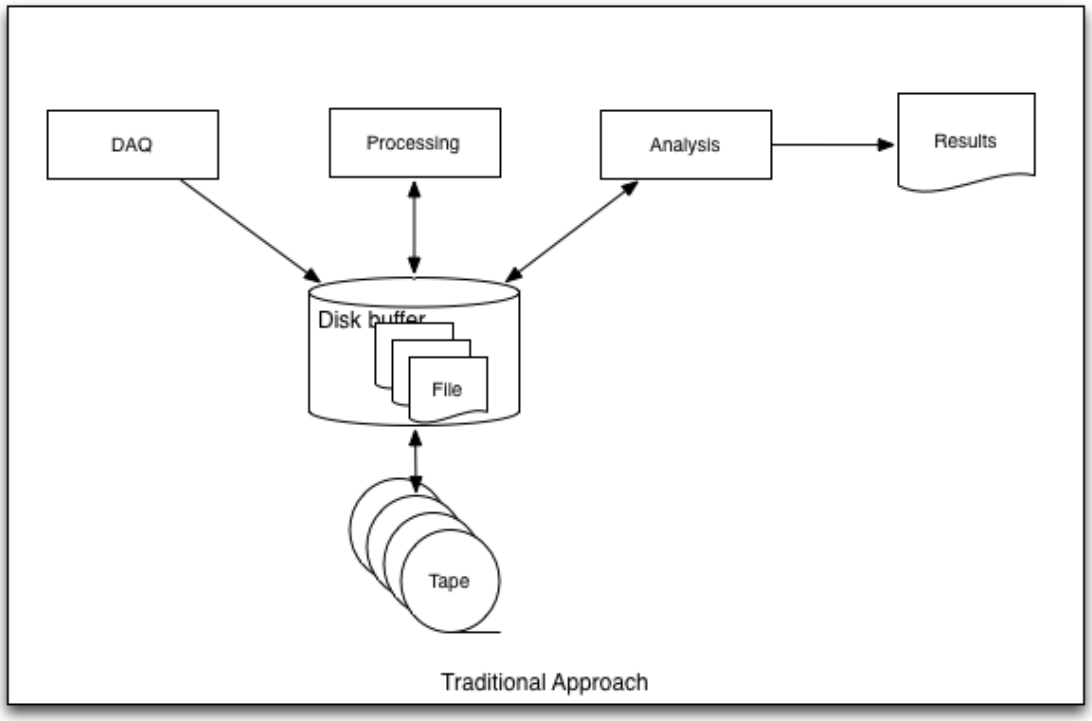
This document does not discuss storing and using of conditions data.

Traditional Model

Traditional data collection, reconstruction and analysis model of HEP data handling is based on the idea of using files to represent event data. File is a unit of operation of the traditional approach. Depending on the experiment, a file may contain single or multiple events. Additional files can be produced during the data processing and analysis. Association between files and events is usually recorded inside the file and/or in some sort of metadata database.

1. Data collection
    a. Receive raw data from the detector
    b. Write raw data to disk as a file
    c. Store files on tape as soon as possible to prevent data loss
2. Reconstruction
    a. read data from file (retrieve from tape if necessary)
    b. reconstruct event information
    c. write results of the reconstruction to disk
    d. associate reconstructed data with raw data in some sort of bookkeeping database or append reconstructed data to the raw file
    e. Store reconstructed data to tape
3. Analysis

a. Retrieve reconstructed (and sometimes raw) data from tape, store in disk buffer
b. Filter "interesting" events
c. Calculate statistics (histograms, cross sections, etc.)
d. Go back to step (b)
e. Store selected "interesting" events for future analysis



Traditional approach uses tape as a permanent storage of data and disk as a fast but limited buffer used for data processing and analysis. Disk is not considered to be reliable.

Disadvantages of the traditional approach:

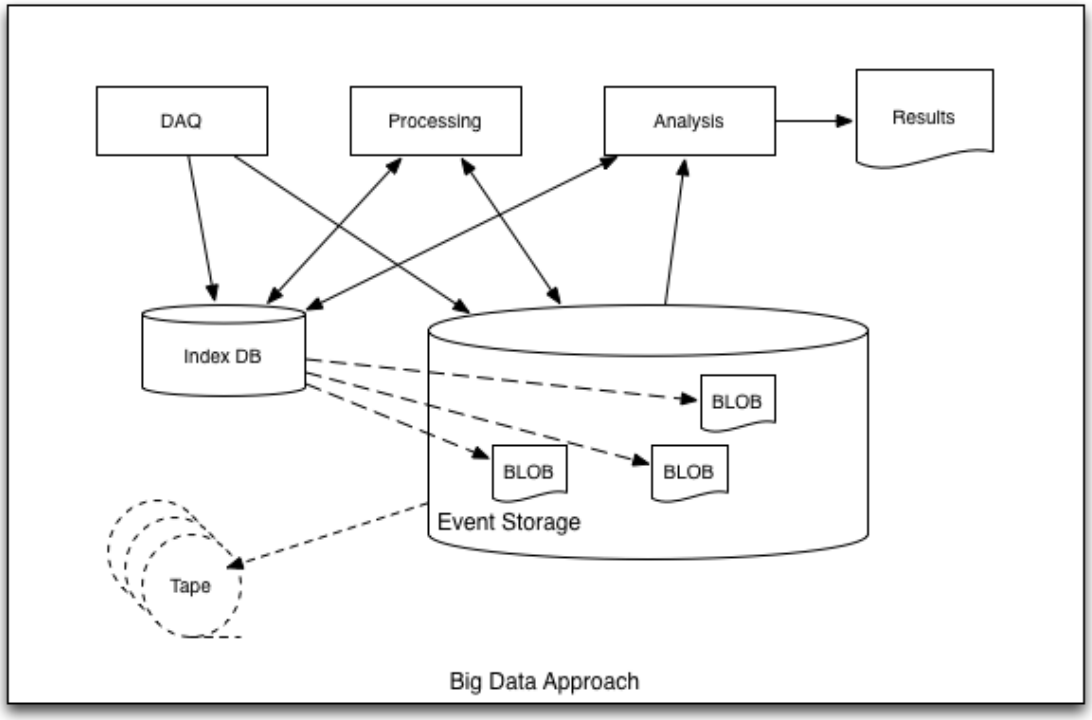
- Data processing efficiency is limited by the disk buffer size
- While unit of information is an event, unit of representation is a file, which translates to the necessity to maintain association between events and files, while information about single event is spread over multiple files
- Often, in order to filter "interesting" events, it is necessary to read multiple files. That is why during the analysis stage, every user or a physics group produces their own sets of files with "interesting" events, where event information is duplicated many times by many different groups or users.

Proposed Big Data Approach

The philosophy of the Big Data approach is to make all data immediately and conveniently available to data processing and analysis.

Immediately means that the data should be stored in fast, direct access storage. Conveniently means that the data is easy to look up and filter by application and user defined criteria.

Big Data approach uses disk for all data storage. Disk storage is distributed among multiple computers combined into one or more clusters.



Big Data Approach

There are 2 storages with different organization and discipline: Event Storage (ES) and Index Database (IDB)

Event Storage (ES)

Event Storage is a no-SQL database (in a very general term) where event information is represented as one or more BLOBs identified by event ID. Event data is stored in the BLOB in application-specific way,

and the ES does not need to know anything about structure of the BLOB.

Event data is never removed from the Event Storage, unless it is considered to be inaccurate or no longer needed. Event Storage is the primary data storage. That is the fundamental difference between the traditional and Big Data approach.

Each BLOB is replicated several (3-5) times within the EventStorage. Data replication serves two purposes: data backup mechanism and access load distribution mechanism. Depending on the data integrity requirements, data should be distributed over several clusters, located at physically different location.

Data in ES can be compressed to reduce space and data transfer throughput requirements.

To provide remote access to data, ES can have a data access interface, which can be used by remote clients to facilitate distributed data processing, for example, via web service/HTTP.

Tape is no longer considered as a primary data storage media. Tape still can be used as a write-once-read-almost-never backup media, but there is no longer need to store data to tape as soon as possible.

Index Database (IDB)

There is also Index Database (IDB). IDB is a (relational) database, which stores information about events. The purpose of IDB is to allow data searching using user defined criteria. It includes DAQ-related event attributes such as event timestamp and trigger configuration, reconstruction status, and even some reconstructed information, which is easy to represent in a relational database. Also, it stores (private or public) attributes defined by physics analysis groups (PAG) or individual users. It also may store fully reconstructed event information such as the tree of 4-momentums the particles produced during the event.

IDB is used during all stages of data processing, from raw data collection to final physical analysis. Important feature of IDB is that it is easily extendable by the collaboration, PAGs or even individuals to accommodate their needs to make the data easily searchable.

To provide remote access to data, IDB can have a data access interface, which can be used by remote clients to facilitate distributed data processing, for example, via web service/HTTP.

Here is new data processing approach:

1. Data collection
    a. Receive raw data from the detector
    b. Write raw data as BLOB to the Event Storage under newly generated event ID. Event Storage will replicate data several times almost immediately.
    c. Raw data indexing - record DAQ information about the event in IDB
2. Reconstruction
    a. Find "new" raw event IDs in the ES using IDB (e.g. using event timestamp)
    b. Retrieve raw data BLOB from the ES by their IDs.
    c. Reconstruct each event
    d. Store reconstructed events into ES under same event IDs
    e. Index reconstructed data - write to IDB
3. Analysis
    a. Using IDB, find event IDs of all "interesting" events
    b. Create your own indexes in IDB
    c. Record index information about "interesting" events in IDB
    d. Go to (a)